Optimization Methods Applied to Aerodynamic Flow Control

Jérémie Labroquère, Régis Duvigneau

European Congress on Computational Methods
in Applied Sciences and Engineering (ECCOMAS 2012)
J. Eberhardsteiner et.al. (eds.)
Vienna, Austria, September 10-14, 2012

OPTIMIZATION METHODS APPLIED TO AERODYNAMIC FLOW CONTROL

Jérémie Labroquère[1], **Régis Duvigneau**[1]

[1]INRIA Sophia Antipolis Méditerranée - OPALE Project-Team
INRIA - 2004, route des Lucioles
BP 93 - 06902 Sophia Antipolis Cedex
e-mail: {jeremie.labroquere, regis.duvigneau}@inria.fr

Keywords: flow control, optimization, aerodynamics

Abstract. *This study deals with the use of optimization algorithms to determine efficient parameters of flow control devices. To improve the performance of systems characterized by detached flows and vortex shedding, the use of flow control devices such as oscillatory jets, are intensively studied, using numerical as well as experimental methods. However, the determination of efficient control parameters is still a bottleneck for industrial problems. Therefore, we propose to couple a global optimization algorithm with an unsteady flow simulation to derive efficient flow control rules.*

We consider as testcase the turbulent flow over a backward facing step, including a synthetic jet actuator. The aim is to reduce the time-averaged recirculation length behind the step by optimizing the jet blowing/suction amplitude and frequency. The Unsteady Reynolds-Averaged Navier-Stokes (URANS) equations are solved within a Mixed finite-Element/finite-Volume (MEV) framework using the near-wall low-Reynolds number one-equation Spalart-Allmaras turbulence closure. The steady flow simulation without control is first validated by comparison with experimental and numerical data. Then, the optimization method EGO (Efficient Global Optimization), based on the construction of a Gaussian surrogate model, is coupled with the solver and applied to the unsteady flow with actuation. It is shown that the time-averaged recirculation length can be shortened when suitable control parameters are used.

1 INTRODUCTION

Shape optimization methods have significantly matured for the last years and automated design loops based on CFD software are now commonly used in industry. However, shape optimization may not be efficient to improve the performance, when the flow is characterized by a strong unsteadiness related to a massive detachement. This is typically the case for the flow around a car or a wing in stall condition. To overcome this difficulty, flow control strategies have been recently developed, that aim at manipulating vortex dynamics by introducing some active actuators, such as periodic blowing/suction jets. In this context, the choice of the control parameters (location, amplitude, frequency) is critical and not straightforward. The design of flow control devices is usually carried out using systematic parametric studies, either on the basis of numerical simulations or experimental campaigns.

Therefore, the objective of this work is to extend some methodologies, developed successfully for shape optimization, to improve the efficiency of flow control devices. The main difficulty is related to the fact that the flows are strongly unsteady. As consequences, the computational cost of each simulation is huge, the objective functional can be highly multimodal and the computation of functional gradients is cumbersome using an adjoint approach.

To overcome these difficulties, a deterministic global optimization method based on surrogate models is used in this study: we extend the approach proposed in [6] for laminar flows, to an unsteady turbulent flow control problem.

2 NUMERICAL FRAMEWORK

We describe in this section the numerical framework used to solve compressible turbulent flow equations. The multidisciplinary platform NUM3SIS [12], developed at INRIA, is used for the implementation. We present the numerical methods employed, and more particularly the implementation of the synthetic jet boundary conditions.

2.1 Mean turbulent flow modeling

The equation solved for the turbulent compressible flow simulations are the Unsteady Favre-Averaged Navier-Stokes equations. For any variable x, with $\overline{x}$ the mean part and x' the fluctuations of x, the Reynolds average or filter is denoted by $x = \overline{x} + x'$ with $x' = 0$. The density weighted average, called Favre averaging or filtering, is denoted by $\widetilde{x} = \overline{\rho x}/\overline{\rho}$ with the related fluctuation part $x'' = x - \widetilde{x}$.

In conservative form, the averaged turbulent compressible equations can be written as:

$$\partial_t \boldsymbol{W^c} + \boldsymbol{\nabla} \cdot \boldsymbol{\mathcal{F}}(\boldsymbol{W^c}) = \boldsymbol{\nabla} \cdot \boldsymbol{\mathcal{N}}(\boldsymbol{W^c}) + \boldsymbol{\mathcal{S}}(\boldsymbol{W^c}) \tag{1}$$

with the averaged conservative variables $\boldsymbol{W^c} = (\overline{\rho}, \overline{\rho}\widetilde{u}, \overline{\rho}\widetilde{v}, \overline{\rho}\widetilde{w}, \overline{\rho}\widetilde{E})^T$, the inviscid flux $\boldsymbol{\mathcal{F}}(\boldsymbol{W^c}) = (\boldsymbol{F}(\boldsymbol{W^c}), \boldsymbol{G}(\boldsymbol{W^c}), \boldsymbol{H}(\boldsymbol{W^c}))^T$, the viscous flux $\boldsymbol{\mathcal{N}}(\boldsymbol{W^c}) = (\boldsymbol{R}(\boldsymbol{W^c}), \boldsymbol{S}(\boldsymbol{W^c}), \boldsymbol{T}(\boldsymbol{W^c}))^T$ and the source term $\boldsymbol{\mathcal{S}}(\boldsymbol{W^c})$.

The components of the inviscid flux in the global frame $\mathcal{R}_0(\widehat{\boldsymbol{x}}, \widehat{\boldsymbol{y}}, \widehat{\boldsymbol{z}})$ are:

$$\boldsymbol{F}(\boldsymbol{W^c}) = \begin{pmatrix} \overline{\rho}\widetilde{u} \\ \overline{\rho}\widetilde{u}^2 + \overline{p} \\ \overline{\rho}\widetilde{u}\widetilde{v} \\ \overline{\rho}\widetilde{u}\widetilde{w} \\ \widetilde{u}(\overline{\rho}\widetilde{E} + \overline{p}) \end{pmatrix}, \quad \boldsymbol{G}(\boldsymbol{W^c}) = \begin{pmatrix} \overline{\rho}\widetilde{v} \\ \overline{\rho}\widetilde{v}\widetilde{u} \\ \overline{\rho}\widetilde{v}^2 + \overline{p} \\ \overline{\rho}\widetilde{v}\widetilde{w} \\ \widetilde{v}(\overline{\rho}\widetilde{E} + \overline{p}) \end{pmatrix}$$

$$\boldsymbol{H}(\boldsymbol{W^c}) = \begin{pmatrix} \overline{\rho}\widetilde{w} \\ \overline{\rho}\widetilde{w}\widetilde{u} \\ \overline{\rho}\widetilde{w}\widetilde{v} \\ \overline{\rho}\widetilde{w}^2 + \overline{p} \\ \widetilde{w}(\overline{\rho}\widetilde{E} + \overline{p}) \end{pmatrix}$$

The components of the viscous flux are:

$$\boldsymbol{R}(\boldsymbol{W^c}) = \begin{pmatrix} 0 \\ \overline{\tau_{xx}} \\ \overline{\tau_{xy}} \\ \overline{\tau_{xz}} \\ \widetilde{u}\overline{\tau_{xx}} + \widetilde{v}\overline{\tau_{xy}} + \widetilde{w}\overline{\tau_{xz}} - \overline{q_x} \end{pmatrix}, \quad \boldsymbol{S}(\boldsymbol{W^c}) = \begin{pmatrix} 0 \\ \overline{\tau_{yx}} \\ \overline{\tau_{yy}} \\ \overline{\tau_{yz}} \\ \widetilde{u}\overline{\tau_{yx}} + \widetilde{v}\overline{\tau_{yy}} + \widetilde{w}\overline{\tau_{yz}} - \overline{q_y} \end{pmatrix}$$

$$\boldsymbol{T}(\boldsymbol{W^c}) = \begin{pmatrix} 0 \\ \overline{\tau_{zx}} \\ \overline{\tau_{zy}} \\ \overline{\tau_{zz}} \\ \widetilde{u}\overline{\tau_{zx}} + \widetilde{v}\overline{\tau_{zy}} + \widetilde{w}\overline{\tau_{zz}} - \overline{q_z} \end{pmatrix}$$

By averaging the Navier-Stokes equations, other quantities as the molecular diffusion $\overline{\boldsymbol{\tau^{lam}} \cdot \boldsymbol{V''}}$, the turbulent transport term $\overline{\rho(\boldsymbol{V''} \cdot \boldsymbol{V''})\boldsymbol{V''}}$ and the turbulent kinetic energy $k = \frac{1}{2}\frac{\overline{\rho \boldsymbol{V''} \cdot \boldsymbol{V''}}}{\overline{\rho}}$ appears but are not shown here. Theses quantities are neglected in this study.

The averaged shear stress tensor $\overline{\boldsymbol{\tau}}$ is made of a laminar and turbulent tensor: $\overline{\boldsymbol{\tau}} = \overline{\boldsymbol{\tau^{lam}}} + \overline{\boldsymbol{\tau^{turb}}}$. The laminar part is modeled considering the flow as beeing Newtonian:

$$\overline{\boldsymbol{\tau}^{lam}} = \overline{\mu_l}\left(\boldsymbol{\nabla}\widetilde{\boldsymbol{V}} + \boldsymbol{\nabla}\widetilde{\boldsymbol{V}}^T - \frac{2}{3}(\boldsymbol{\nabla} \cdot \widetilde{\boldsymbol{V}})\boldsymbol{I_n}\right)$$

The turbulent part uses the Boussinesq eddy-viscosity assumption :

$$\overline{\boldsymbol{\tau}^{turb}} = -\overline{\rho \boldsymbol{V''} \otimes \boldsymbol{V''}} \approx \overline{\mu_t}\left(\boldsymbol{\nabla}\widetilde{\boldsymbol{V}} + \boldsymbol{\nabla}\widetilde{\boldsymbol{V}}^T - \frac{2}{3}(\boldsymbol{\nabla} \cdot \widetilde{\boldsymbol{V}})\boldsymbol{I_n}\right)$$

As for the shear stress tensor, the heat flux $\boldsymbol{q}$ is split in two parts: $\overline{\boldsymbol{q}} = \overline{\boldsymbol{q}^{lam}} + \overline{\boldsymbol{q}^{turb}}$. Both the laminar and turbulent parts are modeled by using a Fick or Fourier law:

$$\boldsymbol{q^{lam}}(\epsilon, \kappa) = -\gamma\frac{\overline{\mu_l}}{\mathrm{Pr}}\boldsymbol{\nabla}\widetilde{\epsilon}$$

$$\boldsymbol{q^{turb}}(\epsilon, \kappa) = \overline{\rho \boldsymbol{V''} h''} \approx -\gamma\frac{\overline{\mu_t}}{\mathrm{Pr_t}}\boldsymbol{\nabla}\widetilde{\epsilon}$$

with $\widetilde{\epsilon} = \frac{\overline{p}}{(\gamma-1)\overline{\rho}}$ and the specific enthalpy $h = \mathrm{c_p}T$ where $\mathrm{c_p}$ is the specific coefficient at constant pressure and T the temperature.

Finally, to close the equations, the total energy is modeled by the perfect gas state law $\widetilde{E} = \widetilde{\epsilon} + \frac{1}{2}\widetilde{\boldsymbol{V}} \cdot \widetilde{\boldsymbol{V}}$, the adiabatic index is set to $\gamma = 1.4$, the Prandtl number is set to $\mathrm{Pr} = 0.72$, the turbulent Prandtl number is set to $\mathrm{Pr_t} = 0.9$ and the laminar viscosity μ_l is modeled by the Sutherland law. The turbulent viscosity $\overline{\mu_t}$ is found by solving the Spalart-Allmaras equation.

For simplicity purpose, the average notations $\bar{\cdot}$ and $\widetilde{\cdot}$ are ommited in the following.

2.2 Spalart-Allmaras turbulent viscosity modeling

The turbulence model used in this study is the basic version of the one-equation Spalart-Allmaras model, where the turbulent viscosity is a function of the variable $\tilde{\nu}$:

$$\mu_t = f_{v1}\rho\max(\tilde{\nu}, 0)$$

with $f_{v1} = \frac{\chi^3}{\chi^3+C_{v1}^3}$, $\chi = \frac{\tilde{\nu}}{\mu_l/\rho}$ and C_{v1} a model constant defined below.

As for the mean equations, the Spalart-Allmaras equation in conservative form can be written as:

$$\partial_t \boldsymbol{W_t^c} + \boldsymbol{\nabla}\cdot\boldsymbol{\mathcal{F}_t}(\boldsymbol{W^c}, \boldsymbol{W_t^c}) = \boldsymbol{\nabla}\cdot\boldsymbol{\mathcal{N}_t}(\boldsymbol{W^c}, \boldsymbol{W_t^c}) + \boldsymbol{\mathcal{S}_t}(\boldsymbol{W^c}, \boldsymbol{W_t^c}) \tag{2}$$

with the Spalart-Allamars conservative variable $\boldsymbol{W_t^c} = (\rho\tilde{\nu})$, the inviscid turbulence flux $\boldsymbol{\mathcal{F}_t}(\boldsymbol{W^c}, \boldsymbol{W_t^c}) = (\rho\tilde{\nu}u, \rho\tilde{\nu}v, \rho\tilde{\nu}w)^T$, the viscous flux $\boldsymbol{\mathcal{N}_t}(\boldsymbol{W^c}, \boldsymbol{W_t^c}) = \mu_e^t\boldsymbol{\nabla}\tilde{\nu}$ with $\mu_e^t = \frac{\mu_l+\rho\tilde{\nu}}{\sigma_m}$, σ_m a model constant and the source flux $\boldsymbol{\mathcal{S}_t}(\boldsymbol{W^c}, \boldsymbol{W_t^c})$.

The components of the turbulence source flux in the global frame $\mathcal{R}_0(\widehat{\boldsymbol{x}}, \widehat{\boldsymbol{y}}, \widehat{\boldsymbol{z}})$ are:

$$\boldsymbol{\mathcal{S}_t}(\boldsymbol{W^c}, \boldsymbol{W_t^c}) = P - D + d$$

with P, D and d respectively the production, destruction and diffusion terms defined by:

$$P = c_{b1}\rho\tilde{\nu}\tilde{S}, \quad D = c_{w1}f_w\rho\left(\frac{\tilde{\nu}}{d_{wall}}\right)^2, \quad d = \frac{c_{b2}}{\sigma_m}\rho\|\boldsymbol{\nabla}\tilde{\nu}_t\|_2^2$$

with $\tilde{S} = \|\boldsymbol{w}\|_2 + \frac{\tilde{\nu}}{\kappa^2 d_{wall}^2}f_{v2}$ limited to be positive, d_{wall} the distance to the wall and the vorticity $\boldsymbol{w}$ defined as $\boldsymbol{w} = \boldsymbol{\nabla}\times\boldsymbol{V}$.

The missing model variables are set to:

$$f_{v1} = \frac{\chi^3}{\chi^3+C_{v1}^3}, \quad f_{v2} = 1 - \frac{\chi}{1+\chi f_{v1}}, \quad f_w = g\left[\frac{1+c_{w3}^6}{g^6+c_{w3}^6}\right]^{1/6}$$

$$g = r + c_{w2}(r^6 - r), \quad r = \min\left[\frac{\tilde{\nu}}{\tilde{S}\kappa^2 d_{wall}^2}, 10\right], \quad \chi = \frac{\tilde{\nu}}{\mu_l/\rho}$$

$\kappa = 0.41$, $c_{b1} = 0.1355$, $\sigma_m = 2/3$, $c_{b2} = 0.622$, $\kappa = 0.41$, $c_{w2} = 0.3$, $c_{w3} = 2$, $c_{v1} = 7.1$, $c_{w1} = \frac{c_{b1}}{\kappa^2} + \frac{1+c_{b2}}{\sigma_m}$.

2.3 Spatial discretization

A Mixed finite-Element/finite-Volume (MEV) discretization is used with a vertex centered approach. The inviscid fluxes and source fluxes are discretized with a finite-volume approach while the viscous fluxes are discretized with a finite-element approach [7, 8].

A polygonal bounded domain $\Omega \subset \mathbb{R}^n$ is considered with a boundary Γ, sub-divided into a tetrahedrization or triangulation $\mathcal{T}_h$ with elements T_i. Around each vertex s_i a finite-volume control cell C_i of a measure $m(C_i)$ is constructed. The set of vertices which are joined to the vertex s_i is denoted by $\mathcal{N}(s_i)$. The subset of all the highest topological dimension polygons sharing the vertex s_i is denoted by $\mathcal{T}(s_i)$.

The inviscid fluxes are computed on the dual control cells C_i while the viscous fluxes are computed on the elements T_i. A weak formulation of the Navier-Stokes equations can be expressed using a Galerkin approach.

By integrating the equation (1) over a control cell S_i (dual control cell or element) against a test function φ_i, the weak formulation can be written as:

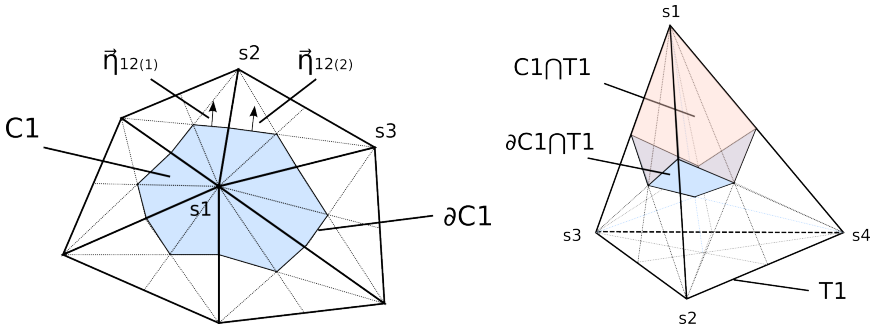
$$\int_{S_i} (\partial_t \boldsymbol{W^c} + \boldsymbol{\nabla} \cdot \boldsymbol{\mathcal{F}}(\boldsymbol{W^c}))\varphi_i d\Omega = \int_{S_i} (\boldsymbol{\nabla} \cdot \boldsymbol{\mathcal{N}}(\boldsymbol{W^c}))\varphi_i d\Omega + \int_{S_i} (\boldsymbol{\mathcal{S}}(\boldsymbol{W^c}))\varphi_i d\Omega \tag{3}$$

2.3.1 Convective fluxes discretization

The finite-volume method can be interpreted as a Galerking method using the control cell $S_i = C_i$ and a test function defined as:

$$\varphi_i^C(\boldsymbol{x}) = \begin{cases} 1 & \text{if } \boldsymbol{x} \text{ is in } C_i \\ 0 & \text{else} \end{cases}$$

The variables $\boldsymbol{W^c}$ are supposed to be constant on each control cell C_i and denoted $\boldsymbol{W_i^c}$ (see Fig. 1(a) and Fig. 1(b)).



(a) A control cell in 2D. (b) A control cell in 3D.

Figure 1: Illustration of control cells.

By using the Green-Ostrogradski theorem, the left-hand side of equation (3) yields:

$$\int_{C_i} (\partial_t \boldsymbol{W^c} + \boldsymbol{\nabla} \cdot \boldsymbol{\mathcal{F}}(\boldsymbol{W^c}))\varphi_i^C d\Omega = m(C_i)\partial_t \boldsymbol{W_i^c} + \int_{\partial C_i} (\boldsymbol{\mathcal{F}}(\boldsymbol{W^c}) \cdot \widehat{\boldsymbol{\eta}}))d\sigma \tag{4}$$

with ∂C_i the boundary of the cell C_i and $\widehat{\boldsymbol{\eta}}$ the outward unit normal vector.

Furthermore, as the domain is discretized,

$$\begin{aligned} \int_{\partial C_i} (\boldsymbol{\mathcal{F}}(\boldsymbol{W^c}) \cdot \widehat{\boldsymbol{\eta}}))d\sigma &= \sum_{s_j \in \mathcal{N}(s_i)} \int_{\partial C_i \cap \partial C_j} (\boldsymbol{\mathcal{F}}(\boldsymbol{W^c}) \cdot \widehat{\boldsymbol{\eta}}))d\sigma \\ &\approx \sum_{s_j \in \mathcal{N}(s_i)} \boldsymbol{\mathcal{F}}(\boldsymbol{W^c})_{\vert ij} \cdot \int_{\partial C_i \cap \partial C_j} \widehat{\boldsymbol{\eta}} d\sigma \\ &= \sum_{s_j \in \mathcal{N}(s_i)} \boldsymbol{\mathcal{F}}(\boldsymbol{W^c})_{\vert ij} \cdot \boldsymbol{\eta_{ij}} \end{aligned}$$

The $\boldsymbol{\mathcal{F}}(\boldsymbol{W^c})_{\vert ij}$ term is estimated by using an approximate Riemann solver for which the associated numerical flux is $\boldsymbol{\Phi_{ij}} = \boldsymbol{\Phi}(\boldsymbol{W_i^c}, \boldsymbol{W_j^c}, \boldsymbol{\eta_{ij}})$. In the current work, for the mean flow

equations, the HLLC [18] numerical flux is used for the internal domain and a modified version of the Steger-Warming flux is used for boundaries [7]. For the estimation of the inviscid flux of the turbulence equation, the density flux computed for the mean flow equations is used [13].

To obtain a high-order approximation in space, a MUSCL reconstruction technique is used. The reconstructed primitive state at the common interface of the cells C_i and C_j is denoted by $\boldsymbol{W_{ij}^p} = \boldsymbol{W_i^p} + \frac{1}{2}\alpha_{ij}(\boldsymbol{\nabla W_i^p}) \cdot \boldsymbol{IJ}$. The slope $(\boldsymbol{\nabla W_i^p}) \cdot \boldsymbol{IJ}$ is approximated by $\Delta \boldsymbol{W_i^p} = 2/3\Delta_{|N}\boldsymbol{W_i^p} + 1/3\Delta_{|C}\boldsymbol{W_{ij}^p}$. The nodal slope $\Delta_{|N}\boldsymbol{W_i^p}$ is the weighted average of the $\mathcal{P}_1$-Galerkin gradients computed on each element $T \in \mathcal{T}(s_i)$. The slope $\Delta_{|C}\boldsymbol{W_{ij}^p}$ corresponds to the centered slope $\boldsymbol{W_j^p} - \boldsymbol{W_i^p}$. We denote by $\alpha_{ij}(\Delta_{|N}\boldsymbol{W_i^p}, \Delta_{|C}\boldsymbol{W_{ij}^p})$ the limiting coefficient. Finally, the high-order numerical flux is computed using extrapolated states $\boldsymbol{\Phi_{ij}}^{\text{High order}} = \boldsymbol{\Phi}(\boldsymbol{W_{ij}^c}, \boldsymbol{W_{ji}^c}, \boldsymbol{\eta_{ij}})$.

2.3.2 Viscous terms discretization

The finite-element method is retrieved by using the control cell $S_i = T_i$. The polygon T_i is defined as a Lagrange $\mathcal{P}_1$ finite-element with the canonical basis functions denoted φ_i^T for each vertex s_i. Thus, φ_i^T is equal to unity at the vertex s_i and zero at the other vertices of the element. The $\mathcal{P}_1$ approximation of any function f on a polygon T_i can be obtained by :

$$f(\boldsymbol{X}) \approx f_h^{T_i}(\boldsymbol{X}) = \sum_{i=1}^{N_s \in T_i} f_i \varphi_i^T(\boldsymbol{X})$$

where f_i is the value of f at the vertex s_i.

By integrating by part and using the Green-Ostrogradski theorem, the diffusive term of equation (3) becomes:

$$\int_{T_i} (\boldsymbol{\nabla} \cdot \boldsymbol{\mathcal{N}}(\boldsymbol{W^c}))\varphi_i^T d\Omega = -\int_{T_i} \boldsymbol{\mathcal{N}}(\boldsymbol{W^c}) \cdot \boldsymbol{\nabla}\varphi_i^T d\Omega + \int_{\partial T_i} (\boldsymbol{\mathcal{N}}(\boldsymbol{W^c}) \cdot \boldsymbol{n})\varphi_i^T d\sigma \tag{5}$$

The velocity gradient $\boldsymbol{\mathcal{N}}$ and $\boldsymbol{\nabla}\varphi_i^T$ being assumed to be constant by element :

$$\int_{T_i} \boldsymbol{\mathcal{N}}(\boldsymbol{W^c}) \cdot \boldsymbol{\nabla}\varphi_i^T d\Omega = \boldsymbol{\mathcal{N}}(T_i) \cdot \boldsymbol{\nabla}\varphi_i^T \int_{T_i} d\Omega = \boldsymbol{\mathcal{N}}(T_i) \cdot \boldsymbol{\nabla}\varphi_i^T m(T_i)$$

By denoting $\boldsymbol{\eta_{T_i}} = -\boldsymbol{\nabla}\varphi_i^T m(T_i)$, the viscous numerical flux is finally defined as:

$$\boldsymbol{\Upsilon_{T_i}} = \boldsymbol{\mathcal{N}}(T_i) \cdot \boldsymbol{\eta_{T_i}}$$

Thus, the equation (5) reads:

$$\int_{T_i} (\boldsymbol{\nabla} \cdot \boldsymbol{\mathcal{N}}(\boldsymbol{W^c}))\varphi_i^T d\Omega = \boldsymbol{\Upsilon_{T_i}} + \int_{\partial T_i} (\boldsymbol{\mathcal{N}}(\boldsymbol{W^c}) \cdot \boldsymbol{n})\varphi_i^T d\sigma \tag{6}$$

The term $\int_{\partial T_i} (\boldsymbol{\mathcal{N}}(\boldsymbol{W^c}) \cdot \boldsymbol{n})\varphi_i^T d\sigma$ concerns the boundary conditions.

2.3.3 Source terms discretization

The source terms are discretized in a finite volume way, by considering the variables are constant on the control cell C_i. The source term of 3 becomes:

$$\int_{C_i} (\boldsymbol{\mathcal{S}}(\boldsymbol{W^c}))\varphi_i^C d\Omega = m(C_i)\boldsymbol{\mathcal{S}}(\boldsymbol{W_i^c})$$

If the source terms require gradient values, a weighted average of the $\mathcal{P}_1$-Galerkin gradients is used, as for the inviscid fluxes MUSCL reconstruction.

2.4 Time integration

An implicit second-order time discretization is obtained by using a dual time step approach [9] and a backward time integration. We introduce the computed residual $\boldsymbol{\mathcal{R}_i}$ for the control cell i. By denoting $\delta_1^2\Lambda = \Lambda^2 - \Lambda^1$ the variation of the variable Λ from the state 1 to the state 2, we obtain as second-order implicit time integration scheme:

$$m(C_i)\frac{3\boldsymbol{W_i^{c^{n+1}}} - 4\boldsymbol{W_i^{c^n}} + \boldsymbol{W_i^{c^{n-1}}}}{2\delta_n^{n+1}t} + \boldsymbol{\mathcal{R}_i}(\boldsymbol{W^{c^{n+1}}}) = 0 \tag{7}$$

To solve this problem, we introduce a sub-iteration state of index k, such as:

$$\boldsymbol{W_i^{c^{n+1}}} = \lim_{k\to\infty} \boldsymbol{W_i^{c^{(k+1)}}}$$

The linearization of equation (7) around the state of index k, with a local time step $\delta_k^{k+1}t_i$ yields:

$$\left(\left(\frac{m(C_i)}{\delta_k^{k+1}t_i} + \frac{3m(C_i)}{2\delta_n^{n+1}t}\right)\boldsymbol{I_n} + \boldsymbol{\mathcal{J}^*}(\boldsymbol{W^{c^{(k)}}})\right)\delta_k^{k+1}\boldsymbol{W_i^c} = -\boldsymbol{\mathcal{R}_i}(\boldsymbol{W^{c^{(k)}}}) + m(C_i)\frac{\delta_{n-1}^n\boldsymbol{W_i^c} - 3\delta_n^k\boldsymbol{W_i^c}}{2\delta_n^{n+1}t}$$

with $\boldsymbol{\mathcal{J}^*}(\boldsymbol{W^{c^{(k)}}})$ an approximate Jacobian of the numerical residuals.

The approximate Jacobian is composed of three parts arising from the inviscid fluxes, the viscous terms and the source terms. The Jacobian of the inviscid fluxes is based on the first-order Rusanov flux. The use of the Rusanov flux, or spectral radius Jacobian approximation, is usually used in matrix-free approaches [11]. The Jacobian of the viscous terms is based on an exact linearization, as in [7]. Concerning the source terms, for the turbulence variables, the full source Jacobian is computed.

The resulting Jacobian matrix is inversed with the Lower-Upper Symmetric Gauss-Seidel (LU-SGS) iterative algorithm. The use of a delta form for the implicit formulation allows to employ an approximated Jacobian without loosing the second order accuracy in space reached with the MUSCL extrapolation.

The time integration of the mean flow equations and the Spalart-Allmaras equation is carried out using a weak coupling. Thus, the turbulence variable is frozen during the resolution of the system for mean flow variables, whereas the system for the turbulence variable is solved with frozen mean flow variables.

2.5 Boundary conditions

2.5.1 Weak approach for inlet and outlet conditions

Some boundary conditions are implemented in a weak form through numerical fluxes, as illustrated in the Fig. 2.

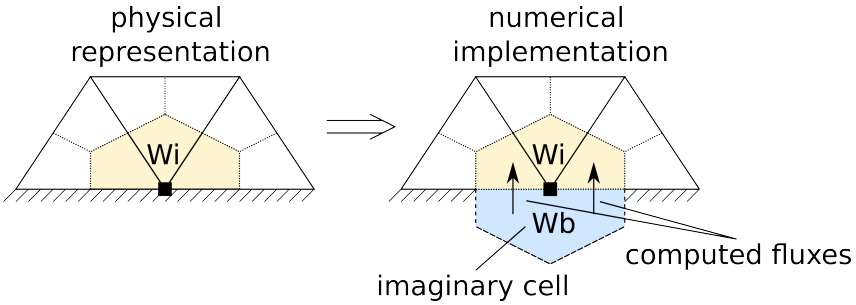


Figure 2: Illustration of the implementation of weak boundary conditions.

The flux estimated using the fictive state $\boldsymbol{W}_{\boldsymbol{b}}^{\boldsymbol{c}}$ and the interior state $\boldsymbol{W}_{\boldsymbol{i}}^{\boldsymbol{c}}$ can be computed with any numerical flux (independently from the interior domain). Depending on the boundary condition used, the fictive state $\boldsymbol{W}_{\boldsymbol{b}}^{\boldsymbol{c}}$ can be a function of the interior $\boldsymbol{W}_{\boldsymbol{i}}^{\boldsymbol{c}}$ and/or the exterior $\boldsymbol{W}_{\boldsymbol{o}}^{\boldsymbol{c}}$ states : $\boldsymbol{W}_{\boldsymbol{b}}^{\boldsymbol{c}} = \boldsymbol{W}_{\boldsymbol{b}}^{\boldsymbol{c}}(\boldsymbol{W}_{\boldsymbol{i}}^{\boldsymbol{c}}, \boldsymbol{W}_{\boldsymbol{o}}^{\boldsymbol{c}})$. In this study, the computation of the fictive state is carried out using imposed density and velocity with extrapolation of the pressure for subsonic inlets, imposed pressure with extrapolation of characteristics for subsonic outlets (characteristic-based methods), and symmetry conditions for slipping walls. Furthermore, for all the inlet/outlet boundaries, a non-reflexive version of the Steger-Warming flux is used [7], allowing to correctly filter the entering or exiting characteristics.

2.5.2 Strong approach for non-slip condition

The strong boundary condition approach consists in imposing the condition at the boundary and modifying the Jacobian and the residuals in such a way that the boundary condition is verified implicitly when solving the linear system. This boundary condition is used when the value of one particular conservative variable is completely defined at the boundary. The interior state $\boldsymbol{W}_{\boldsymbol{i}}^{\boldsymbol{c}}$ is set to be equal to the fictive state $\boldsymbol{W}_{\boldsymbol{b}}^{\boldsymbol{c}}$. The resulting flux computed at the boundary is a physical flux as shown on the Fig. 3.

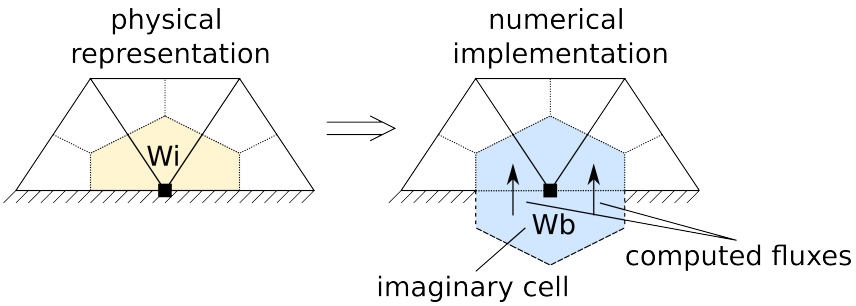


Figure 3: Illustration of the implementation of strong boundary conditions.

Thus, the non-slip boundary condition is imposed by setting at the boundary a zero momentum for the mean flow equations and a zero turbulent variable for the Spalart-Allmaras equation.

2.5.3 Iterative approach for jet condition

The iterative boundary conditions are implemented in the cases where it is cumbersome to use strong boundary conditions. This can occur for instance when the user would like to impose a primitive variable while the Jacobian and residuals are expressed using conservative variables. We use the dual time step approach to implement iteratively the boundary condition, in order to reach a target $\boldsymbol{W_i^c}^{target}$ at the step $n+1$, at the boundary :

$$\begin{cases} \frac{m(C_i)}{\delta_n^{n+1} t_i} \delta_n^{n+1} \boldsymbol{W_i^c} + \boldsymbol{\mathcal{R}_i}^{n+1}(\boldsymbol{W_i^c}^n) = 0 \\ \lim\limits_{k \to \infty} \delta_n^{n,k+1} \boldsymbol{W_i^c} = \delta_n^{n+1} \boldsymbol{W_i^c} = \delta_n^{target} \boldsymbol{W_i^c} \end{cases}$$

In other words, this boundary condition consists in imposing at each dual time step sub-iterations the targeted value of the variable. As described in [1], the internal fields and the boundary conditions converge towards the solution with the targeted boundary condition at the next physical step.

The jet boundary conditions are implemented using a mixed iterative/weak approach. In particular, the velocity profile is set with the iterative boundary condition while the other variables are imposed using a weak form.

The jet velocity profile is modeled by:

$$\boldsymbol{V_j} = A(\mathbf{x}) \sin(\omega t + \phi)\ \boldsymbol{d_j}$$

with A(**x**) a profile function, ω the angular frequency of the jet, ϕ the phase and $\boldsymbol{d_j}$ the direction of the jet. In this study, $A(\mathbf{x})$ is a sine squared function. The other variables are extrapolated from the interior of the domain.

3 OPTIMISATION METHODS

3.1 Kriging model of the flow response

We use a kriging model to describe the variation of the objective function value, for instance the drag coefficient or the recalculation length, with respect to control parameters. Kriging models (also called Gaussian process models) belong to response surface methods, that allow to predict a function value f at a given point x, on the basis of a set of known function values $F_N = \{f_1, f_2, \ldots, f_N\}$ at some points $X_N = \{x_1, x_2, \ldots, x_N\} \subset \mathbb{R}^d$, that are stored in a database.

Kriging models [14, 17] treat the response of some experiments as if it were a realization of a stochastic process, because the model relies on a finite number of observations and is therefore subject to uncertainties. In the following, we adopt the Bayesian viewpoint of Gaussian processes [20]. The vector of known function values F_N is assumed to be one realization of a multivariate Gaussian process with joint probability density:

$$p(F_N) = \frac{\exp\left(-\frac{1}{2} F_N^\top C_N^{-1} F_N\right)}{\sqrt{(2\pi)^N \det(C_N)}}, \tag{8}$$

where C_N is the $N \times N$ covariance matrix. The element C_{mn} of the covariance matrix gives the correlation between the function values f_m and f_n obtained respectively at points x_m and

x_n. We assume that these values are correlated, since they correspond to underlying physical phenomena. This is expressed in terms of a correlation function c, i.e., $C_{mn} = c(x_m, x_n)$.

Now, we suppose that we would like to evaluate the function value at a new point x_{N+1}. When adding a new point x_{N+1}, the resulting vector of function values F_{N+1} is assumed to be a realization of the $(N+1)$-variable Gaussian process with joint probability density:

$$p(F_{N+1}) = \frac{\exp\left(-\frac{1}{2}F_{N+1}^\top C_{N+1}^{-1} F_{N+1}\right)}{\sqrt{(2\pi)^{N+1}\det(C_{N+1})}}. \tag{9}$$

Using the rule of conditional probabilities $p(A|B) = p(A,B)/p(B)$, we can write the probability density for the unknown function value f_{N+1}, given the data (F_N) as:

$$p(f_{N+1}|F_N) = \frac{p(F_{N+1})}{p(F_N)}. \tag{10}$$

In order to simplify this expression we notice that the $(N+1)$-variable covariance matrix C_{N+1} can be written as

$$C_{N+1} = \begin{bmatrix} C_N & k \\ k^\top & \kappa, \end{bmatrix}$$

where

$$k = [c(x_1, x_{N+1}), c(x_2, x_{N+1}), \ldots, c(x_N, x_{N+1})]^\top \quad \text{and} \quad \kappa = c(x_{N+1}, x_{N+1}).$$

This allows us to write the inverse of correlation matrix as

$$C_{N+1}^{-1} = \begin{bmatrix} M & m \\ m^\top & \mu, \end{bmatrix}$$

where

$$M = C_N^{-1} + \frac{1}{\mu} m m^\top, \qquad m = -\mu C_N^{-1} k, \qquad \mu = (\kappa - k^\top C_N^{-1} k)^{-1}.$$

Then, we obtain that the probability density for the function value at the new point is

$$p(f_{N+1}|F_N) \propto \exp\left[-\frac{(f_{N+1} - \hat{f}_{N+1})^2}{2\sigma_{f_{N+1}}^2}\right],$$

where

$$\hat{f}_{N+1} = k^\top C_N^{-1} F_N \qquad \sigma_{f_{N+1}}^2 = \kappa - k^\top C_N^{-1} k. \tag{11}$$

Thus, the probability density for the function value at the new point x_{N+1} is also Gaussian with mean $\hat{f}_{N+1}$ and standard deviation $\sigma_{f_{N+1}}$. Therefore, the most likely value at the new point x_{N+1} is $\hat{f}_{N+1}$. This value will be considered as the prediction of the kriging model. The variance $\sigma_{f_{N+1}}$ can be interpreted as a measure of uncertainty in the value prediction. The function value can be expected to vary in some range like $[\hat{f} - 3\sigma, \hat{f} + 3\sigma]$.

The covariance function must reflect the characteristics of the output of the computer code since, it represents the way the different point values are correlated. In the absence of any knowledge regarding the unknown function, the most commonly used correlation function is an exponential; in this work we take the correlation function of the form:

$$c(x, y) = \theta_1 \exp\left[-\frac{1}{2}\sum_{i=1}^{d} \frac{(x_i - y_i)^2}{r_i^2}\right] + \theta_2,$$

where $\Theta = (\theta_1, \theta_2, r_1, r_2, \ldots, r_d)$ are some parameters to be determined. The first term is a distance-dependent correlation between the function values at two data points; if their distance is small compared to the length scales r_i, the exponential term is close to one while it decays exponentially as their distance increases. Practical studies have shown that the use of a non-isotropic scaling is beneficial for the problems considered here. The parameter θ_1 scales this correlation. In the second term, θ_2 gives an offset of the function values from zero.

It now remains to find the parameters Θ in the correlation function. These parameters are determined by maximizing the joint probability density $p(F_N)$. Indeed, we want the constructed model to be most consistent with the observed data. This is equivalent to minimizing the log-likelihood function given by:

$$\mathfrak{L} = F_N^\top C_N^{-1} F_N + \log\det(C_N)$$

This function is known to be multi-modal; hence an evolution strategy is employed for this work, that has the capability to avoid local minima. There are many practical issues that must be taken care of so that all the computations are stable; we follow [3] in this respect. The hyper-parameters must be non-negative; hence it is better to work with the logarithm of the parameters so that they always remain positive. The correlation matrix can be ill-conditioned in which case the computation of its inverse will not be accurate. If the condition number is above a specified tolerance, then the log-likelihood is taken to be a large positive value. If the condition number is within the specified tolerance, an LU decomposition of C_N is first performed; then $C_N^{-1} F_N$ and $C_N^{-1} k$ are computed using the LU decomposition. The logarithm of the determinant of C_N is also computed using the LU decomposition

$$\log\det(C_N) = \sum_{n=1}^{N} \log L_{nn} + \sum_{n=1}^{N} \log U_{nn}$$

This avoids the under-flow problem associated with multiplying a large number of small numbers which would be the case if the matrix C_N is badly scaled. An upper limit of $1/\epsilon$ is imposed on the condition number of C_N, where ϵ is the machine precision.

Again following [3] we scale the coordinates and function values. Moreover, the parameters in the covariance function are constrained by upper and lower bounds:

$$\begin{aligned} \theta_1 &\in [10^{-3}, 1] \\ \theta_2 &\in [10^{-3}, 1] \\ r_i &\in [10^{-2}, 10], \quad i = 1, \ldots, d \end{aligned}$$

3.2 Optimization strategy based on kriging models

The optimization strategy used in this study is based on the iterative construction of a kriging model (see [10] for a review paper on these methods). The use of such a model for optimization must be an iterative process since it is not possible to build a model accurate enough to find the optimal parameters in only one step. The model should be updated with the results from flow simulations until some convergence criterion is fulfilled. Therefore, the algorithm is organized in two phases. During the first one, an initial *a priori* database is constructed, that gathers the flow responses (objective function values) corresponding to different control parameter values. The control parameters are chosen in order to explore uniformly the search space, according to a Design Of Experiments (DOE) method. During the second phase, a kriging model is constructed on the basis of available data and is used to determine which flow simulations

should be carried out and added into the database. This second phase is then repeated until convergence of the algorithm.

Using such an algorithm, the database and the model are iteratively enriched and reconstructed. The choice of the optimization algorithm used to solve the minimization problem in step 3 is not critical, since this is an inexpensive task. An evolution strategy is used in practice since it is robust and not sensitive to local optima.

The robustness and efficiency of this algorithm depends critically on the choice of the new simulations to be carried out during the step 2. To obtain satisfactory results, the capability of a kriging model to predict the uncertainty related to value predictions is used. Indeed, for a given design vector x, a kriging model provides not only a prediction of the function value $\hat{f}(x)$, but also an estimation of the uncertainty of this prediction $\sigma(x)$. Therefore, one can use this knowledge to determine not only the point that minimizes the model, but also areas for which the model is uncertain, i.e. of poor quality. Evaluating such points promotes the minimization of the cost function as well as the improvement of the kriging model.

Thus, we define a *merit function* that can be interpreted as a statistical lower bound of the model, by aggregating the function value and the standard deviation value [10]:

$$f_M(x) = \hat{f}(x) - \rho\sigma(x), \tag{12}$$

where ρ is a positive parameter that allows to balance the two terms. The algorithm can finally be summarized as:

1. Build an *a priori* database
2. Construct a kriging model
3. For $i = 1$ to p :
   Find the point $x_i^\star$ that minimizes the merit function $\hat{f}(x) - \rho_i\sigma(x)$
   (with $\rho_i = i - 1$ in general)
4. Evaluate the p points $(x_i^\star)_{i=1,\dots,p}$ and add them in the database
5. Return to step 2 until convergence

The use of $\rho_1 = 0$ yields the minimization of the kriging model. Moreover, the use of other values, e.g. $\rho_2 = 1$, $\rho_3 = 2$, promotes the exploration of areas where the standard deviation is high and where possible interesting points can be found. These extra points will be useful since they allow to improve the model quality. This approach has been found robust and applied to various problems [3, 5, 15].

The optimization process is stopped according to three possible criteria : the user can set a maximum number of simulations to perform or a target for the cost function value. Moreover, there is a third criterion, related to the iterative construction of the kriging model: the process is stopped when the minimization of the merit functions yields to points already in the database (for a given accuracy).

Nevertheless, this algorithm exhibits some limitations: first, the kriging model can be sensitive to the error originating from the evaluation process of the cost function. Typically, if the cost function is evaluated through a numerical simulation, a low convergence of the solver can lead the search to spurious local optima. Secondly, this approach is limited to problems of

rather low dimension. The number of parameters should be typically lower than twenty in order to have an efficient search, according to the literature results. However, this is usually the case for flow control applications. In a previous work [15], this algorithm was used to solve aerodynamic shape optimization problems on the basis of inviscid flow models, with an increasing number of design parameters. In this context, it has been found that the proposed algorithm provides satisfactory results up to 32 variables.

4 APPLICATION TO FLOW CONTROL ON A BACKWARD FACING STEP

The backward facing step has been selected to test the optimization method on a flow control problem. This test-case has been widely used for code and turbulence models validation [16]. Despite the fact that the turbulence model and the jet model used in this study are quite simple, this problem will demonstrate the capability to optimize control parameters for detached flows.

4.1 Computational conditions

The backward facing step configuration and flow conditions correspond to the case presented by Driver and Seegmiller in [4]. The boundary conditions used for the computation are shown on the Fig. 4. The inlet flow Mach number is set to $M = 0.128$ and the velocity magnitude is set to $U_{ref} = 44.2$ m/s. The outlet pressure is set to $P = 1.0015\ P_\infty$. The Reynolds number based on the boundary layer thickness is 5000. In our computations, the top wall is replaced by a symmetry boundary condition. The first point of the synthetic jet is located at $-h/50$ from the step corner. Its diameter is set to $h/10$ and its direction is perpendicular to the wall. The amplitude and frequency ranges are defined in section 4.1.2.

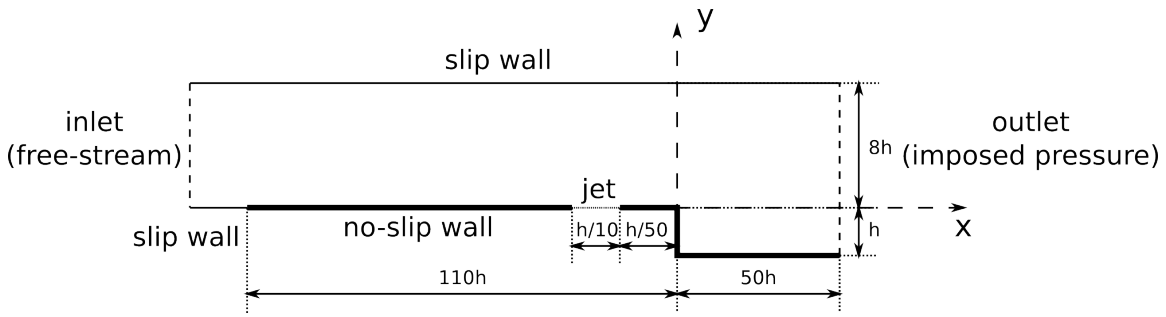


Figure 4: Illustration of the backward facing step configuration.

The unsteady computation is initialized using a steady solution, corresponding to the same case with an inactive jet. Calculation is performed with a time step of $t* = tV_{ref}/h = 0.004$. The dual time step sub-iterations are stopped when they have reached 15 iterations or when the residual has been reduced of 5 orders.

4.1.1 Computational grid

All the grid generated are unstructured grids on which control cells are constructed following Barth method [2, 19].

The steady computation is done on three different grids. From coarse to fine, the grids contains respectively 15946, 29606 and 61344 vertices. For each grid, the distance between the first interior node and the wall is 5.10^{-6}. For all the cases, this restriction on the grid

size ensures to fulfill the condition $y+ < 1$. Presently, the unsteady computations and the optimization exercise are only done on the coarse grid.

4.1.2 Design of experiment

The optimization parameters considered in this study are restricted to the amplitude and frequency of the synthetic jet. To construct the *a priori* database used to explore the design space, these jet parameters are supposed to vary in the following ranges:

- Amplitude: $0.1 \leq U_{jet}/U_{ref} \leq 1.0$
- Frequency: $0.05 \leq fh/U_{ref} \leq 0.5$

Then, an initial sampling of 16 flows is generated using an optimized Latin Hypercube Sampling (LHS) , as illustrated in figure 5.

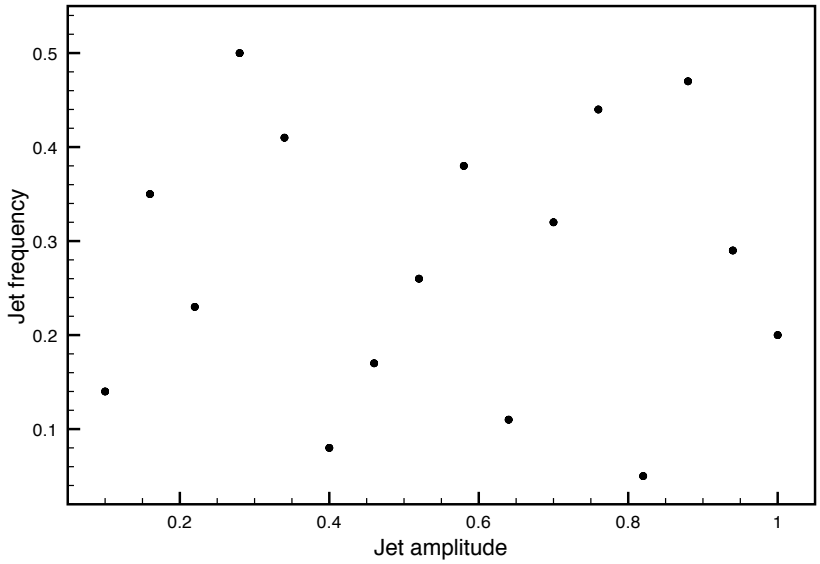
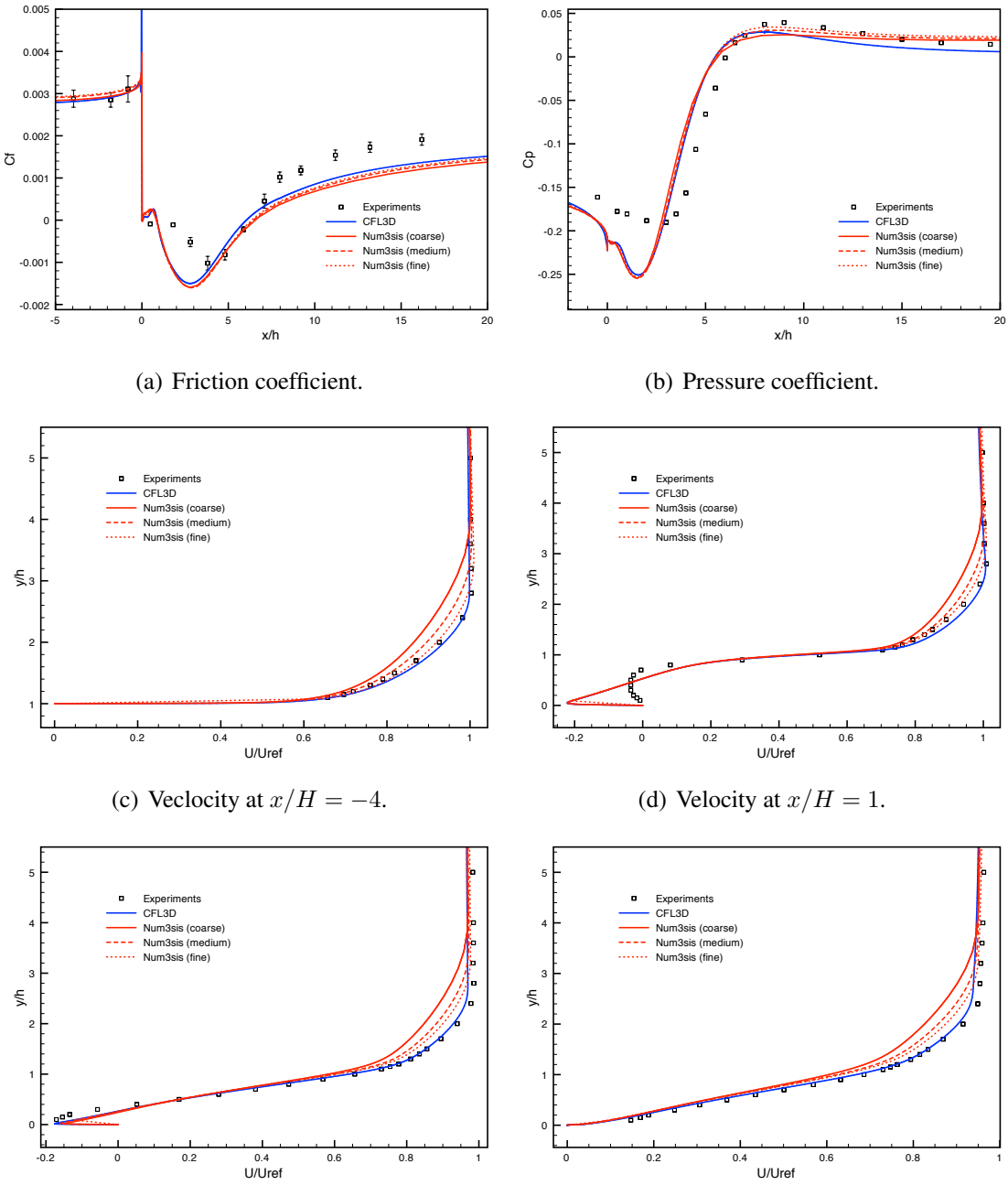


Figure 5: Illustration of the initial sampling generated using LHS.

4.2 Results and discussion

4.2.1 Validation without actuation

We first validate our computations by considering the case without actuation. The case is the same as the backward facing step configuration presented in 4.1, but with a wall instead of the jet boundary condition. The friction coefficient, the pressure and velocity distibution profiles obtained with the NUM3SIS platform are compared to experimental data and numerical results from the CFL3D code developed at NASA [4, 16]. As seen in figure 6, the results converge to those obtained by the CLF3D code with a very fine grid (about 300 000 nodes). However, one can notice that the Spalart-Allmaras model is not able to reproduce correctly the flow in the recirculation area. Nevertheless, the length of the recirculation bubble and the reattachment point are quite well predicted, even on the coarsest grid. Therefore, the coarsest grid is used in the following for the unsteady computations.



(a) Friction coefficient.

(b) Pressure coefficient.

(c) Veclocity at $x/H = -4$.

(d) Velocity at $x/H = 1$.

(e) Velocity at $x/H = 4$.

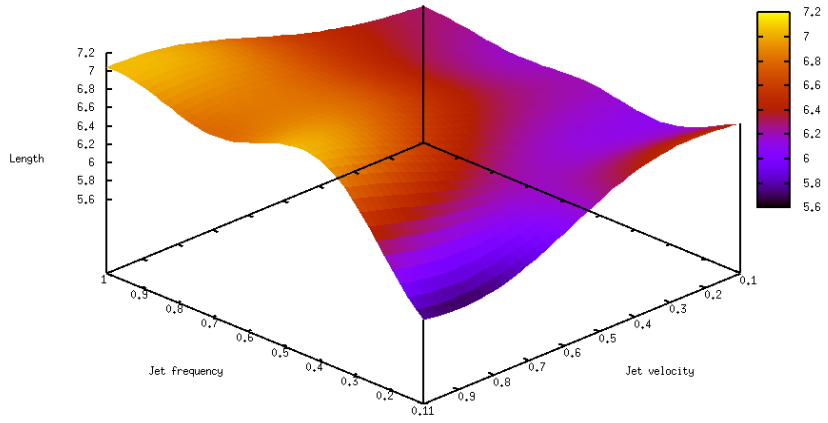
(f) Velocity at $x/H = 6$.

Figure 6: Comparison of NUM3SIS and CFL3D on the steady case.

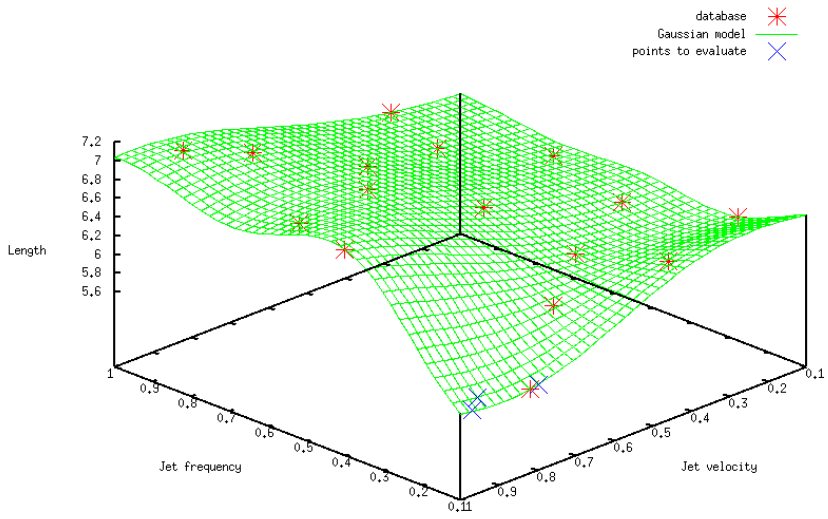
4.2.2 Surrogate model construction

Based on the LHS, 16 different actuations are simulated using NUM3SIS. The time-averaged recirculation lenght l/h computed at the wall is selected to be the function of interest. A kriging model is constructed and can be seen in figure 7(a). Using such a model, one can easily identify some parameter regions for which the length of the recirculation bubble is reduced. According to the algorithm described above, three points are then selected by minimization of the three merit functions. These new candidates can be seen in figure 7(b). Then, these new points have

to be simulated and included into the database. Unfortunately, these computations are still in progress.



(a) Surrogate model of the function of interest.



(b) Database and new points to be evaluated.

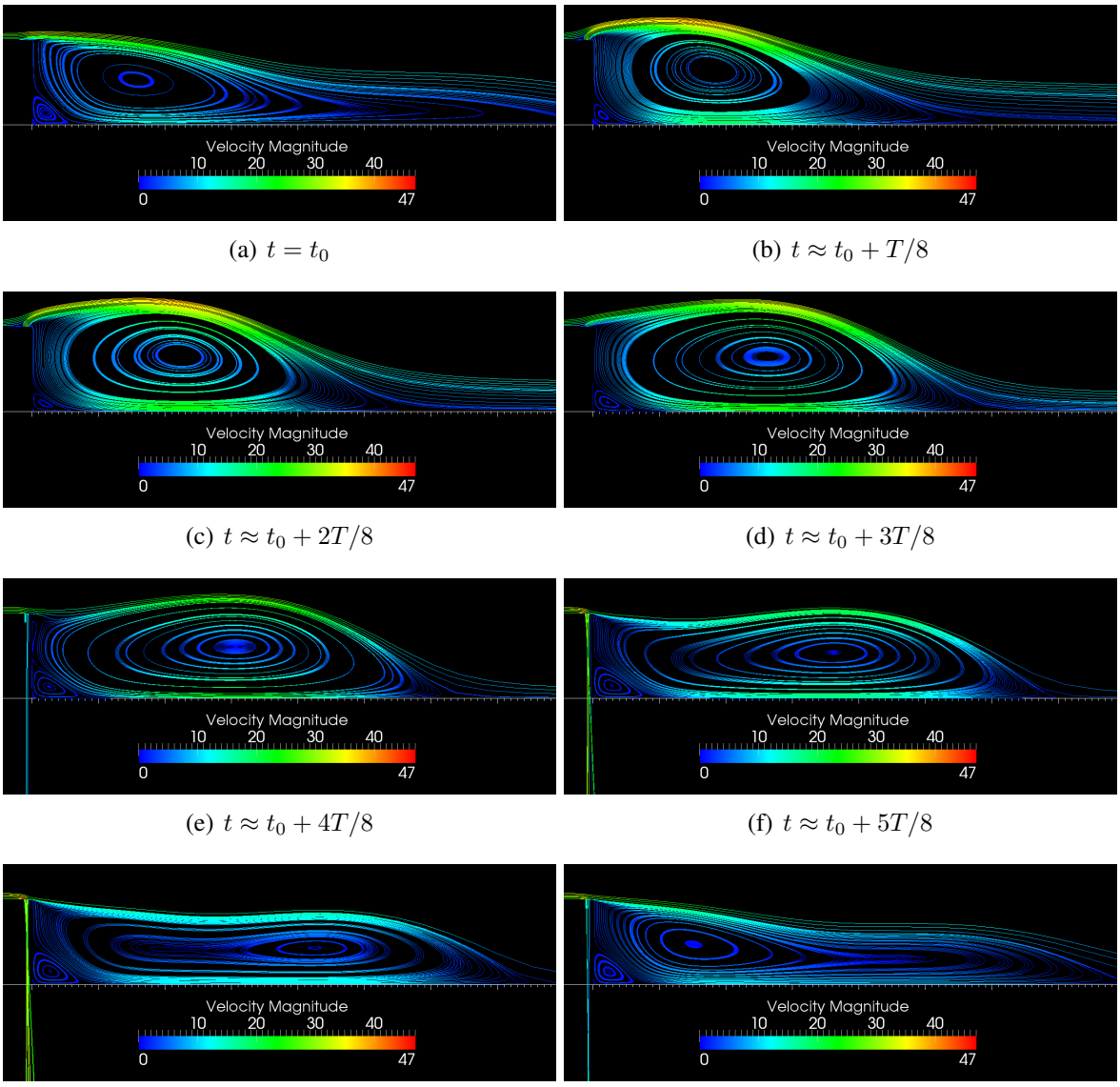
Figure 7: Surrogate model of recirculation length.

4.2.3 Controlled case

Although the optimization process is not finished, the kriging model has already determined a parameter region for which the actuation allows to reduce the time-averaged length of the recirculation bubble. A suitable set of parameters to achieve this result is typically :

- Large blowing amplitude: $U_{jet}/U_{ref} \approx 0.8$
- Low frequency actuation: $F_{jet}h/U_{ref} \approx 0.05$

With this configuration, the figure 8 provides some snapshots of the flow along an actuation period. As can be seen, a large recirculation bubble is generated by the actuation, which makes the reattachment point moving. Figure 9 shows a comparison between the stationary flow without actuation and the time-averaged flow with actuation. In particular, one can observe a reduction of the time-averaged recirculation length of about 5%. One can also notice the change of the shape of the reticulation bubble, whose thickness is higher in the case of actuation. Better results might be obtained by moving the jet or changing the jet direction.



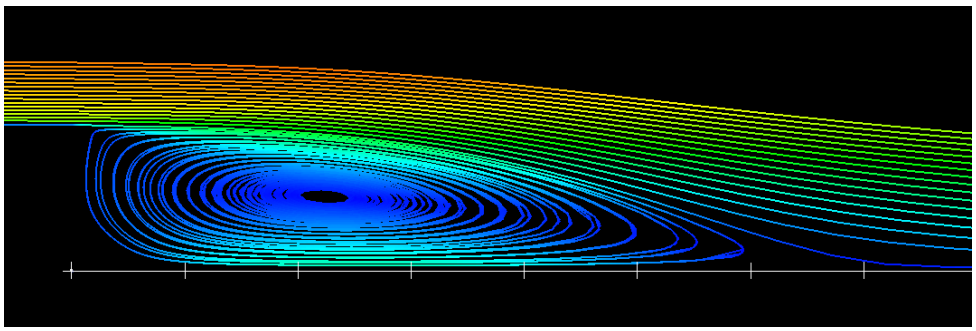
(a) $t = t_0$ (b) $t \approx t_0 + T/8$

(c) $t \approx t_0 + 2T/8$ (d) $t \approx t_0 + 3T/8$

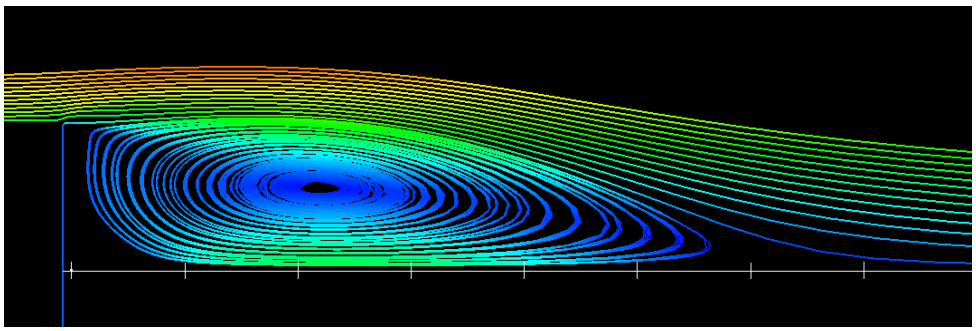
(e) $t \approx t_0 + 4T/8$ (f) $t \approx t_0 + 5T/8$

(g) $t \approx t_0 + 6T/8$ (h) $t \approx t_0 + 7T/8$

Figure 8: Streamlines over a cycle for the large blowing amplitude and low frequency actuation.



(a) Reference case



(b) Averaged controlled case

Figure 9: Streamlines comparison between the reference state and the averaged controlled state.

4.3 Conclusion and prospects

The kriging-based optimization algorithm has been coupled with an unsteady RANS solver. This approach has been applied to optimize the amplitude and frequency of a synthetic jet located just before a backward facing step, in order to shorten the time-averaged recirculation length. This study has shown the capability of the kriging model to determine efficient control parameters, although the complete optimization process has not been achieved.

These results should be then carefully validated by quantifying the influence of numerical parameters, such as grid size and time step, as well as numerical models and in particular by assessing the role of the turbulence closure.

REFERENCES

[1] G. Ashcroft and J. Schulz. Numerical modelling of wake-jet interaction with application to active noise control in turbomachinery. *AIAA journal*, 2004.

[2] T.J. Barth. Aspects of unstructured grids and finite-volume solvers for the euler and navier-stokes equations. In *Lecture Notes Presented at the VKI Lecture Series*, volume 1994, 1995.

[3] D. Buche, N.N. Schraudolph, and P. Koumoutsakos. Accelerating evolutionary algorithms with gaussian process fitness function models. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 35(2):183–194, 2005.

[4] D.M. Driver and H.L. Seegmiller. Features of a reattaching turbulent shear layer in divergent channel flow. *AIAA journal*, 23(2):163–171, 1985.

[5] F. Duchaine, T. Morel, and L.Y.M. Gicquel. Computational-fluid-dynamics-based kriging optimization tool for aeronautical combustion chambers. *AIAA Journal*, 47(3):631–645, 2009.

[6] R. Duvigneau and P. Chandrashekar. Kriging-based optimization applied to flow control. *International Journal for Numerical Methods in Fluids*, 2011.

[7] Loula Fezoui, Fatima, Stephane Lanteri, Bernard Larrouturou, and Christian Olivier. Resolution numerique des equations de Navier-Stokes pour un fluide compressible en maillage triangulaire. Rapport de recherche RR-1033, INRIA, 1989.

[8] J. Francescatto. *Méthodes multigrilles par agglomération dirrectionnelle pour le calcul d'écoulements turbulents*. PhD thesis, Université de Nice-Sophia Antipolis, Nice, France, 1998.

[9] A. Jameson. Time dependent calculations using multigrid, with applications to unsteady flows past airfoils and wings. *AIAA paper*, 91:1596, 1991.

[10] D.R. Jones. A taxonomy of global optimization methods based on response surfaces. *Journal of Global Optimization*, 21(4):345–383, 2001.

[11] T. Kloczko. *Développement d'une méthode implicite sans matrice pour la simulation 2D-3D des écoulements compressibles et faiblement compressibles en maillages non-structurés*. PhD thesis, Arts et Métiers ParisTech, 2006.

[12] J. Labroquère, R. Duvigneau, T. Kloczko, and J. Wintz. Interactive computation and visualization towards a virtual wind tunnel. In *47th 3AF Symposium on Applied Aerodynamics, Paris, France*, March 2012.

[13] Bernard Larrouturou. How to preserve the mass fractions positivity when computing compressible multi-component flows. Rapport de recherche RR-1080, INRIA, 1989.

[14] D.J.C. MacKay. Introduction to gaussian processes. *NATO ASI Series F Computer and Systems Sciences*, 168:133–166, 1998.

[15] C. Praveen and Régis Duvigneau. Study of some strategies for global optimization using Gaussian process models with application to aerodynamic design. Rapport de recherche RR-6964, INRIA, 2009.

[16] C.L. Rumsey and J.L. Thomas. Application of fun3d and cfl3d to the third workshop on cfd uncertainty analysis. In *Proceedings of the 3rd Workshop on CFD Uncertainty Analysis*, 2008.

[17] J. Sacks, W.J. Welch, T.J. Mitchell, and H.P. Wynn. Design and analysis of computer experiments. *Statistical science*, 4(4):409–423, 1989.

[18] EF Toro, M. Spruce, and W. Speares. Restoration of the contact surface in the hll-riemann solver. *Shock waves*, 4(1):25–34, 1994.

[19] C. Viozat, C. Held, K. Mer, and A. Dervieux. On vertex-centered unstructured finite-volume methods for stretched anisotropic triangulations. *Computer methods in applied mechanics and engineering*, 190(35):4733–4766, 2001.

[20] C.K.I. Williams. Prediction with gaussian processes: From linear regression to linear prediction and beyond. *NATO ASI SERIES D BEHAVIOURAL AND SOCIAL SCIENCES*, 89:599–621, 1998.